2016 International Conference on Artificial Intelligence and Computer Science (AICS 2016)
ISBN: 978-1-60595-411-0

A Bi-criterion Simulated Annealing Method for a Single Batch-processing Machine Scheduling Problem with Dynamic Job Release Times

Ya-Chih TSAI(Hui-Mei WANG)[1] and Fuh-Der CHOU[2,*]

[1]Department of Hotel Management, Vanung University, Tao Yuan, Taiwan, R.O.C.

[2]Department of Industrial Management, Chien Hsin University, Tao Yuan, Taiwan, R.O.C

* E-mail: fdchou@uch.edu.tw

Keywords: Single batch-processing machine, Simulated annealing, Sterilization operation, Scheduling.

Abstract. This paper is motivated by a sterilization operation in a professional waste disposal services, and models the operation as the problem of single batch-processing machine (BPM) scheduling with the objectives of minimizing the makespan and the total weighted number of tardy orders that has not been considered in the literature. The considered problem is NP-hard such that two simulated annealing algorithms (SA1 and SA2) are developed in this work. SA1 uses multi-search directions to find Pareto-optimal solutions by eleven combinations of weights, as to SA2 applied the SMOSA of Suman to solve the considered problem. A computational experiment is conducted and the solutions for each instance obtained by SA1 and SA2 respectively are compared. Computational results reveal that SA1 is significantly better than SA2 because SA1 changes different search directions by two parameters.

Introduction

This paper focuses on a sterilization operation found in professional disinfection, sterilization and waste disposal services. The sterilization machine could be modeled as a batch-processing machine. Each job has non-identical job sizes, dictated by size of device and quantities, and a number of jobs can be processed as a batch in the machine as long as the total sizes of jobs do not exceed the capacity of the machine. Once processing of a batch has been started, no jobs can be removed or added to the batch.

Many researchers have addressed BPM problems recently. Mathirajan and Sivakumar [1] have done a quite complete survey on scheduling with BPM. However, there has been little reported research on bi-objective scheduling problem for a single BPM. Liu et al. [2] considered a variety of bicriterion scheduling problems for a static single BPM in which the processing time are all equal. Qi and Tu [3] considered the earliness and tardiness (E/T) minimization problem on a single BPM in which all jobs have the same processing time and proposed a dynamic programming method to solve it. Sung et al. [4] and Mönch et al. [5] considered the same problem by Qi and Tu [3] where all jobs have unit job size. Li et al. [6] extended the problem of Mönch et al. to the case of non-identical job sizes. They addressed heuristic algorithms based on some properties and developed a hybrid genetic algorithm. Kashan et al. [7] attempted minimizing makespan and maximum tardiness simultaneously for a single BPM problem with non-identical job sizes. They proposed two multi-objective genetic algorithms based on different representation schemes.

For the considered BPM problem, we adopted two objective functions of minimizing the makespan and total weighted number of tardy orders due to bi- or multi-objective decision-making is an essence of real-world problems, and these two objectives are respectively represented as machine's throughput and customer's satisfaction, both are equally important to a company. With the inherent complexity of the considered problem, we develop two simulated annealing (SA) methods to find all pareto-optimal solutions as possible because considerable volume of work in single and multiobjective algorithm based on SA has been proposed and solved successfully for NP-hard problems in the last two decades (Suppapitnarm et al. [8], Suresh and Sahu [9]).

Bi-criterion simulated annealing methods

In this paper, we develop two SA algorithms called SA1 and SA2 for the considered problem. In SA1 the search space tries to be covered by changing the weights of makespan and total weighted number of tardy orders. As to SA2, it essentially applied SMOSA proposed by Suppapitnarm et al. [8] for the considered problem. The step-by-step procedure for SA1 algorithm is described as follow, while the procedure of SA2 could referred to the studies of Suppapitnarm et al. [8], and Suman and Kumar [10].

The step-by-step procedure of SA1

Step1: Generated an initial sequence π^0 randomly, and calculate the weighted objective value for the initial sequence π^0 by α and β where $\beta = 1 - \alpha$ and α increases by 0.1, that is, $\alpha = 0.0 + 0.1x$.

Step 1.1 Set the eleven combined weighted objective values as $f_x(\alpha, \beta, \pi_x)$ where x=0, 1,…,10 and $\pi_x = \pi^0$, and the makespan value and the total weighted number of tardy orders as $C_{max}(\pi_x)$ and $\mathrm{WNT}(\pi_x)$, respectively.

Step 1.2 Set the cooling rate, and the initial temperature be equal to 0.98 and $T_x = 1.618 \times f_x(\alpha, \beta, \pi)$, respectively.

Step 2. x=0

Step 3. Do while $x \leq 10$

Generate a neighbor sequence π^{new} from π_x

Maintain Pareto-optimal set by comparing π^{new} with each of Pareto-optimal solution.

Set y=0

Do while y≤10

$\alpha = 0.0 + 0.1\mathrm{y}, \beta = 1 - \alpha$

$f^{new}(\alpha, \beta, \pi^{new}) = \alpha C_{max}(\pi^{new}) + \beta \mathrm{WNT}(\pi^{new})$

IF ($f^{new}(\alpha, \beta, \pi^{new}) < f^{best}(\alpha, \beta, \pi^{best})$)

$f^{best}(\alpha, \beta, \pi^{best}) = f_x(\alpha, \beta, \pi_x) = f^{new}(\alpha, \beta, \pi^{new})$

$C_{max}(\pi^{best}) = C_{max}(\pi_x) = C_{max}(\pi^{new})$,

$\mathrm{WNT}(\pi^{best}) = \mathrm{WNT}(\pi_x) = \mathrm{WNT}(\pi^{new})$

Else

IF($f^{new}(\alpha, \beta, \pi^{new}) < f_x(\alpha, \beta, \pi_x)$)

$f_x(\alpha, \beta, \pi_x) = f^{new}(\alpha, \beta, \pi^{new})$

$C_{max}(\pi_x) = C_{max}(\pi^{new}), \mathrm{WNT}(\pi_x) = \mathrm{WNT}(\pi^{new})$

Else

Generate a random number γ

$\Delta f = f^{new}(\alpha, \beta, \pi^{new}) - f_x(\alpha, \beta, \pi_x)$

$$p = \exp(\frac{\Delta f}{T_x})$$

IF ($\gamma \leq p$)

$f_x(\alpha, \beta, \pi_x) = f^{new}(\alpha, \beta, \pi^{new})$

$C_{max}(\pi_x) = C_{max}(\pi^{new}), \mathrm{WNT}(\pi_x) = \mathrm{WNT}(\pi^{new})$

End IF

End IF

End IF

y=y+1

End Do

$T_x = 0.98T_x$, x=x+1

IF (execute time≥time_limit)

Stop

End IF

End Do.

Results and discussion

We generated a test-bed problems randomly to examine the two SAs. For each job, the release time, processing times and sizes have been generated from a discrete uniform distribution within [0, 24], [8, 15] and [10, 30]. The weights of jobs are within [1, 5]. The due dates of jobs are defined as $d_j = (r_j + p_j) + U[0{,}0.5 \times Max(p_j)]$ where U indicates a uniform distribution within [a, b].

All proposed SA algorithms are implemented in C++, running on a on a PC Xeon E5-1620 CPU with 3.6 GHz and 12 GB memory. Two performance measures are adopted as following:

(1) A quality performance for A(B) can be defined as $|P_{A(B)} \cap P_C| / |P_C|$ where $|P_A|$ and $|P_B|$ denotes the number of Pareto-solutions obtained by algorithms A and B, respectively. Let $|P_C|$ be the number of non-dominated solutions obtained by the enumeration method. (Hyun et al. [11]).

(2) Efficiency of approximating Pareto front (EAPF) addressed by Smutinicki et al. [12]. For a fair comparison, a bounded ratio is calculated by $BA_{A(B)}/BA_C$ where $BA_{A(B)}$ denotes the bounded area calculated by algorithm A and B, respectively. BA_C is the bounded area calculated by non-dominated solutions obtained by the enumeration method.

To be fair for this comparison, the computation time limit is set to $number\ of\ jobs/10$ for both SA1 and SA2.

In Table 1, all Pareto solutions could be found by the enumeration method, SA1 found all Pareto solutions for 45 instances over 50 while SA2 found all Pareto solutions only for 30 instances. In terms of efficiency, SA1 is also better than SA2. This results exhibit that SA1 is significantly better than SA2 due to SA1 could find more non-dominate solutions by changing different search directions while SA2 did not.

Table 1. the performance of SA1 and SA2 for the test-bed with 12 jobs with 3 customers.

No.	quality performance		Efficiency (bounded ratio)			No.	quality performance		Efficiency (bounded ratio)		
	SA1	SA2	SA1	SA2	EM		SA1	SA2	SA1	SA2	EM
1	1.0	1.0	1.000	1.000	1.000	26	1.0	1.0	1.000	1.000	1.000
2	1.0	0.5*	1.000	0.414	1.000	27	1.0	1.0	1.000	1.000	1.000
3	1.0	1.0	1.000	1.000	1.000	28	1.0	1.0	1.000	1.000	1.000
4	1.0	0.7*	1.000	0.875	1.000	29	1.0	1.0	1.000	1.000	1.000
5	1.0	1.0	1.000	1.000	1.000	30	0.5*	0.5*	0.802	0.802	1.000
6	1.0	0.50*	1.000	0.592	1.000	31	0.7*	0.7*	0.794	0.794	1.000
7	1.0	0.7*	1.000	0.954	1.000	32	1.0	1.0	1.000	1.000	1.000
8	1.0	1.0	1.000	1.000	1.000	33	1.0	0.5*	1.000	0.962	1.000
9	1.0	0.5*	1.000	0.674	1.000	34	1.0	1.0	1.000	1.000	1.000
10	1.0	1.0	1.000	1.000	1.000	35	1.0	1.0	1.000	1.000	1.000
11	1.0	1.0	1.000	1.000	1.000	36	1.0	0.5*	1.000	0.774	1.000
12	1.0	0.5*	1.000	0.630	1.000	37	1.0	1.0	1.000	1.000	1.000
13	1.0	0.5*	1.000	0.846	1.000	38	1.0	0.7*	1.000	0.685	1.000
14	0.5*	0.5*	0.890	0.890	1.000	39	1.0	1.0	1.000	1.000	1.000
15	1.0	0.5*	1.000	0.909	1.000	40	1.0	1.0	1.000	1.000	1.000
16	1.0	1.0	1.000	1.000	1.000	41	0.5*	0.5*	0.737	0.737	1.000
17	1.0	1.0	1.000	1.000	1.000	42	1.0	1.0	1.000	1.000	1.000
18	1.0	1.0	1.000	1.000	1.000	43	1.0	0.5*	1.000	0.940	1.000
19	1.0	1.0	1.000	1.000	1.000	44	1.0	1.0	1.000	1.000	1.000
20	1.0	0.0*	1.000	0.771	1.000	45	1.0	1.0	1.000	1.000	1.000
21	1.0	1.0	1.000	1.000	1.000	46	1.0	1.0	1.000	1.000	1.000
22	1.0	1.0	1.000	1.000	1.000	47	0.7*	0.7*	0.910	0.910	0.910
23	1.0	1.0	1.000	1.000	1.000	48	1.0	1.0	1.000	1.000	1.000
24	1.0	1.0	1.000	1.000	1.000	49	1.0	1.0	1.000	1.000	1.000
25	1.0	0.0*	1.000	0.554	1.000	50	1.0	0.5*	1.000	0.970	1.000

Remark: EM means enumeration method

Conclusions

In this paper, we addressed the single BPM scheduling problem with dynamic job release time. This problem is motivated by scheduling of the sterilization operation in a professional waste disposal services, and the objectives of minimizing the makespan and the total weighted number of tardy orders that has not been considered for BPM problems to date. For the considered problem, we proposed two SA algorithms to obtain Pareto-optimal solutions. A test-beds are examined the performances for the proposed SA algorithms, the results reveal that SA1 outperformed SA2 because SA1 could search a large-scale solution space by changing the weights of α and β.

Summary

If you follow the "checklist" your paper will conform to the requirements of the publisher and facilitate a problem-free publication process.

Acknowledgement

This paper was supported in part by the Ministry of Science and Technology, ROC, under the contract MOST 104-2221-E-231-003-MY2.

References

[1] M. Mathirajan and A.I. Sivakumar, A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. International Journal of Advanced Manufacturing Technology, 29 (2006) 990-1001.

[2] L.L., Liu, C.T. Ng, T.C.E. Cheng, Bicriterion scheduling with equal processing times on a batch processing machine, Computers & Operations Research, 36 (2009) 110-118.

[3] X.T. Qi and F.S. Tu, Earliness and tardiness scheduling problems on a batch processor, Discrete Applied Mathematics, 98 (1999) 131-145.

[4] C.S. Sung, Y.I. Choung, J.W. Fowler, Heuristic algorithm for minimizing earliness-tardiness on single burn-in oven in semicounductor manufacturing, In Proceedings of 2002 international conference of modeling and analysis for semiconductor manufacturing (MASM), pp. 217-222, Tempe, Arizona.

[5] L. Mönch, R. Unbehaun, Y.I. Choung, Minimizing earliness-tardiness on a single burn-in oven with a common due date and maximum allowable tardiness constraint, OR Spectrum, 28 (2006) 177-198.

[6] Z. Li, H. Chen, R. Xu, X. Li, Earliness-tardiness minimization on scheduling a batch processing machine with non-identical job sizes, Computer & Industrial Engineering, 87 (2015) 590-599.

[7] A.H. Kashan, B. Karimi, F. Jolai, An effective hybrid multi-objective genetic algorithm for bi-criteria scheduling on a single batch processing machine with non-identical job sizes, Engineering Applications of Artificial Intelligence, 23 (2010) 911-922.

[8] A. Suppapitnarm, K.A., Seffen, G.T. Parks, and P.J. Clarkso, Simulated annealing: an alternative approach to true multi-objective optimization. Engine Optim, 33 () 59.

[9] G. Suresh, S. Sahu, Stochastic assembly line balancing using simulated annealing, International Journal of Production Research, 32 (1994) 1801-1810.

[10] A. Suman, P. Kumar, A survey of simulated annealing as a tool for single and multiobjective optimization, Journal of the Operational Research Society, 57 (2006) 1143-1160.

[11] C.J. Hyun, Y. Kim, Y.K. Kim, A genetic algorithm for multiple objectives sequencing problem in mixed model assembly lines, Computers & Operations Research, 25 (1998) 675-690.

[12] C. Smutnicki, J. Pempera, J. Rudy, D. Zelazny, A new approach for multi-criteria scheduling, Computers & Industrial Engineering, 90 (2015) 212-220.